

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Image Retrieval Systems and Methods with Semantic
and Feature Based Relevance Feedback**

Inventor(s):
Wen-Yin Liu
Hong-Jiang Zhang

ATTORNEY'S DOCKET NO. MS1-604US

1 **TECHNICAL FIELD**

2 This invention relates to image retrieval systems.

3
4 **BACKGROUND**

5 The popularity of digital images is rapidly increasing due to improving
6 digital imaging technologies and easy availability facilitated by the Internet. More
7 and more digital images are becoming available every day.

8 Automatic image retrieval systems provide an efficient way for users to
9 navigate through the growing numbers of available images. Traditional image
10 retrieval systems allow users to retrieve images in one of two ways: (1) keyword-
11 based image retrieval or (2) content-based image retrieval. Keyword-based image
12 retrieval finds images by matching keywords from a user query to keywords that
13 have been manually added to the images. One of the more popular collections of
14 annotated images is "Corel Gallery", an image database from Corel Corporation
15 that includes upwards of 1 million annotated images.

16 One problem with keyword-based image retrieval systems is it can be
17 difficult or impossible for a user to precisely describe the inherent complexity of
18 certain images. As a result, retrieval accuracy can be severely limited because
19 images that cannot be described or can only be described ambiguously will not be
20 retrieved successfully. In addition, due to the enormous burden of manual
21 annotation, there are few databases with annotated images, although this is
22 changing.

23 Content-based image retrieval (CBIR) finds images that are similar to low-
24 level image features of an example image, such as color histogram, texture, shape,
25 and so forth. Although CBIR solves the problem of keyword-based image

1 retrieval, it also has severe shortcomings. One drawback of CBIR is that searches
2 may return entirely irrelevant images that just happen to possess similar features.
3 Additionally, individual objects in images contain a wide variety of low-level
4 features. Therefore, using only the low-level features will not satisfactorily
5 describe what is to be retrieved.

6 To weed out the irrelevant images returned in CBIR, some CBIR-based
7 image retrieval systems utilize user feedback to gain an understanding as to the
8 relevancy of certain images. After an initial query, such systems estimate the
9 user's ideal query by monitoring user-entered positive and negative responses to
10 the images returned from the query. This approach reduces the need for a user to
11 provide accurate initial queries.

12 One type of relevance feedback approach is to estimate ideal query
13 parameters using only the low-level image features. This approach works well if
14 the feature vectors can capture the essence of the query. For example, if the user
15 is searching for an image with complex textures having a particular combination
16 of colors, this query would be extremely difficult to describe but can be reasonably
17 represented by a combination of color and texture features. Therefore, with a few
18 positive and negative examples, the relevance feedback process is able to return
19 reasonably accurate results. On the other hand, if the user is searching for a
20 specific object that cannot be sufficiently represented by combinations of available
21 feature vectors, these relevance feedback systems will not return many relevant
22 results even with a large number of user feedbacks.

23 Some researchers have attempted to apply models used in text information
24 retrieval to image retrieval. One of the most popular models used in text
25 information retrieval is the vector model. The vector model is described in such

1 writings as Buckley and Salton, "Optimization of Relevance Feedback Weights,"
2 in Proc of SIGIR'95; Salton and McGill, "Introduction to Modern Information
3 Retrieval," McGraw-Hill Book Company, 1983; and W.M. Shaw, "Term-
4 Relevance Computation and Perfect Retrieval Performance," Information
5 processing and Management. Various effective retrieval techniques have been
6 developed for this model and many employ relevance feedback.

7 Most of the previous relevance feedback research can be classified into two
8 approaches: query point movement and re-weighting. The query point movement
9 method essentially tries to improve the estimate of an "ideal query point" by
10 moving it towards good example points and away from bad example points. The
11 frequently used technique to iteratively improve this estimation is the Rocchio's
12 formula given below for sets of relevant documents D'_R and non-relevant
13 documents D'_N noted by the user:

14
15
16

$$Q' = \alpha Q + \beta \left(\frac{1}{N_{R'}} \sum_{i \in D'_R} D_i \right) - \gamma \left(\frac{1}{N_{N'}} \sum_{i \in D'_N} D_i \right) \quad (1)$$

17
18 where α , β , and γ are suitable constants and $N_{R'}$ and $N_{N'}$ are the number of
19 documents in D'_R and D'_N respectively. This technique is implemented, for
20 example, in the MARS system, as described in Rui, Y., Huang, T. S., and
21 Mehrotra, S. "Content-Based Image Retrieval with Relevance Feedback in
22 MARS," in Proc. IEEE Int. Conf. on Image proc., 1997.

23 The central idea behind the re-weighting method is very simple and
24 intuitive. Since each image is represented by an N dimensional feature vector, the
25 image may be viewed as a point in an N dimensional space. Therefore, if the

1 variance of the good examples is high along a principle axis j , the values on this
2 axis are most likely not very relevant to the input query and a low weight w_j can be
3 assigned to the axis. Therefore, the inverse of the standard deviation of the j^{th}
4 feature values in the feature matrix is used as the basic idea to update the weight
5 w_j . The MARS system mentioned above implements a slight refinement to the re-
6 weighting method called the standard deviation method.

7 Recently, more computationally robust methods that perform global
8 optimization have been proposed. One such proposal is the MindReader retrieval
9 system described in Ishikawa, Y., Subramanya R., and Faloutsos, C., "Mindreader:
10 Query Databases Through Multiple Examples," In Proc. of the 24th VLDB
11 Conference, (New York), 1998. It formulates a minimization problem on the
12 parameter estimating process. Unlike traditional retrieval systems with a distance
13 function that can be represented by ellipses aligned with the coordinate axis, the
14 MindReader system proposed a distance function that is not necessarily aligned
15 with the coordinate axis. Therefore, it allows for correlations between attributes in
16 addition to different weights on each component.

17 A further improvement over this approach is described in Rui, Y., Huang,
18 T. S. "A Novel Relevance Feedback Technique in Image Retrieval," ACM
19 Multimedia, 1999. Their CBIR system not only formulates the optimization
20 problem but also takes into account the multi-level image model.

21 All the approaches described above perform relevance feedback at the low-
22 level feature vector level in image retrieval, but fail to take into account any
23 semantics for the images themselves. The inherent problem with these approaches
24 is that adopting relevance feedback used in text information retrieval to image
25 retrieval does not prove to be as successful as hoped. This is primarily because

1 low-level features are often not as powerful in representing complete semantic
2 content of images.

3 As a result, there have been efforts on incorporating semantics in relevance
4 feedback for image retrieval. In Lee, Ma, and Zhang, "Information Embedding
5 Based on User's Relevance Feedback for Image Retrieval," Technical Report HP
6 Labs, 1998, the authors propose a framework that attempts to embed semantic
7 information into a low-level feature-based image retrieval process using a
8 correlation matrix. In this framework, semantic relevance between image clusters
9 is learned from a user's feedback and used to improve the retrieval performance.

10 There remains, however, need for improvement in the image retrieval
11 systems and methods that utilize relevance feedback. The inventors propose a
12 system that integrates both semantics and low-level features into the relevance
13 feedback process in a new way. Only when the semantic information is not
14 available is the technique reduced to one of the previously described low-level
15 feedback approaches as a special case.

16 SUMMARY

17
18 An image retrieval system performs both keyword-based and content-based
19 image retrieval. A user interface allows a user to specify a query using a
20 combination of keywords and examples images. Depending on the input query,
21 the image retrieval system finds images with keywords that match the keywords in
22 the query and/or images with similar low-level features, such as color, texture, and
23 shape. The system ranks the images and returns them to the user.

24 The user interface allows the user to identify images that are more relevant
25 to the query, as well as images that are less or not relevant. . The image retrieval

1 system monitors the user feedback and uses it to refine any search efforts and to
2 train itself for future search queries.

3 In the described implementation, the image retrieval system seamlessly
4 integrates feature-based relevance feedback and semantic-based relevance
5 feedback. With feature-based relevance feedback, the system learns which low-
6 level features led to relevant images and groups such features together to aid
7 future searches.

8 With semantic-based relevance feedback, the image retrieval system learns
9 which keywords are identified with the relevant images and strengthens the
10 associations between the keywords and images. More specifically, the images and
11 keywords are maintained in a database and a semantic network is constructed on
12 top of the image database to define associations between the keywords and images
13 and updated when user feedbacks are provided. Weights are assigned to the
14 keyword-image associations to indicate how relevant the keyword is to the image.
15 The weights are adjusted according to the user feedback, thereby strengthening
16 associations between keywords and images identified as more relevant and
17 weakening the associations between keywords and images identified as less
18 relevant.

19 20 **BRIEF DESCRIPTION OF THE DRAWINGS**

21 Fig. 1 is a block diagram of an exemplary computer network in which a
22 server computer implements an image retrieval system that may be accessed over
23 a network by one or more client computers.

24 Fig. 2 is a block diagram of the image retrieval system architecture.
25

1 Fig. 3 illustrates a semantic network that represents relationships between
2 keywords and images.

3 Fig. 4 is a flow diagram of an initial query handling process in which a user
4 initially submits a keyword query for an image.

5 Fig. 5 is a flow diagram of a refinement and learning process in which the
6 image retrieval system learns from the user's feedback pertaining to how relevant
7 the images are to the initial query.

8 Fig. 6 illustrates a first screen view of a user interface for the image
9 retrieval system.

10 Fig. 7 illustrates a second screen view of the user interface for the image
11 retrieval system.

12 13 **DETAILED DESCRIPTION**

14 This disclosure describes an image retrieval system that performs both
15 keyword-based and content-based image retrieval. The system seamlessly
16 integrates feature-based relevance feedback and semantic-based relevance
17 feedback. The image retrieval system also supports a semantic network
18 constructed on top of an image database to associate keywords with images and
19 employs machine learning to adapt the semantic network based on user feedback.

20 The image retrieval architecture is described in the context of an Internet-
21 based system in which a server hosts the image retrieval system and clients submit
22 user queries to the server. However, the architecture may be implemented in other
23 environments. For instance, the image retrieval architecture may be implemented
24 in non-Internet-based client-server systems or on a non-networked computer
25 system.

Exemplary Computing Environment

Fig. 1 shows an exemplary computer network system 100 in which the image retrieval system may be implemented. The network system 100 includes a client computer 102 that submits queries to a server computer 104 via a network 106, such as the Internet. While the image retrieval system can be implemented using other networks (e.g., a wide area network or local area network) and should not be limited to the Internet, the system will be described in the context of the Internet as one suitable implementation. The web-based retrieval system allows multiple users to perform retrieval tasks simultaneously at any given time.

The client 102 is representative of many diverse computer systems, including general-purpose computers (e.g., desktop computer, laptop computer, etc.), network appliances (e.g., set-top box (STB), game console, etc.), and the like. The client 102 includes a processor 110, a volatile memory 112 (e.g., RAM), and a non-volatile memory 114 (e.g., ROM, Flash, hard disk, optical, etc.). The client 102 also has one or more input devices 116 (e.g., keyboard, keypad, mouse, remote control, stylus, microphone, etc.) and a display 118 to display images returned from the image retrieval system.

The client 102 is equipped with a browser 120, which is stored in non-volatile memory 114 and executed on processor 110. The browser 120 submits requests to and receives responses from the server 104 via the network 106. For discussion purposes, the browser 120 may be configured as a conventional Internet browser that is capable of receiving and rendering documents written in a markup language, such as HTML (hypertext markup language). The browser may further be used to present the images on the display 118.

1 The server 104 is representative of many different server environments,
2 including a server for a local area network or wide area network, a backend for
3 such a server, or a Web server. In this latter environment of a Web server, the
4 server 104 may be implemented as one or more computers that are configured
5 with server software to host a site on the Internet 106, such as a Web site for
6 searching.

7 The server 104 has a processor 130, volatile memory 132 (e.g., RAM), and
8 non-volatile memory 134 (e.g., ROM, Flash, hard disk, optical, RAID memory,
9 etc.). The server 104 runs an operating system 136 and an image retrieval system
10 140. For purposes of illustration, operating system 136 and image retrieval system
11 140 are illustrated as discrete blocks stored in the non-volatile memory 134,
12 although it is recognized that such programs and components reside at various
13 times in different storage components of the server 104 and are executed by the
14 processor 130. Generally, these software components are stored in non-volatile
15 memory 134 and from there, are loaded at least partially into the volatile main
16 memory 132 for execution on the processor 130.

17 The image retrieval system 140 searches for images stored in image
18 database 142. The image retrieval system 140 includes a query handler 150, a
19 feature and semantic matcher 152, and a feedback analyzer 154.

20 The query handler 150 handles queries received from the client 102. The
21 queries may be in the form of natural language queries, individual word queries, or
22 image queries that contains low-level features of an example image that forms the
23 basis of the search. Depending on the query type, the query handler 150 initiates a
24 keyword or feature-based search of the image database 142.
25

1 The feature and semantic matcher 152 attempts to find images in image
2 database 142 that contain low-level features resembling the example image and/or
3 have associated keywords that match keywords in the user query. The feature and
4 semantic matcher 152 utilizes a semantic network to locate images with similar
5 keywords. The semantic network defines associations between the keywords and
6 images. Weights are assigned to the associations to indicate how relevant certain
7 keywords are to the images. One exemplary semantic network is described below
8 in more detail with reference to Fig. 3.

9 The feature and semantic matcher 152 rank the images according to their
10 relevance to the query and return the images in rank order for review by the user.
11 Via a user interface, the user can mark or otherwise identify individual images as
12 more relevant to the query or as less or not relevant to the query.

13 The feedback analyzer 154 monitors the user feedback and analyzes which
14 images are deemed relevant to the search and which are not. The feedback
15 analyzer 154 uses the relevance feedback to train the semantic network in the
16 image database. For instance, the feedback analyzer 154 can modify the
17 annotations on relevant images to more closely comply with the keywords in the
18 search query. The analyzer 154 may also adjust the weights of the semantic
19 network by strengthening associations among keywords of the search query and
20 relevant images, and weakening associations among keywords and non-relevant
21 images.

22 Accordingly, the image retrieval system seamlessly integrates content-
23 based image retrieval (CBIR) and semantic-based image retrieval. The system
24 also integrates semantic and feature-based relevance feedback. The system yields
25 tremendous advantages in terms of both retrieval accuracy and ease of use.

Image Retrieval System Architecture

Fig. 2 illustrates the image retrieval system architecture 140 in more detail. It has a user interface (UI) 200 that accepts both text-based keyword or natural language queries and selection of example images. Thus, a user may choose to enter words or select an example image to use as the initial search query. The UI 200 also provides navigation tools to allow the user to browse through multiple images. In the Fig. 1 network system, the UI 200 can be served as an HTML document and rendered on the client display. One exemplary implementation of the user interface 200 is described below in more detail beneath the heading "User Interface".

The query is passed to the query handler 150. In the illustrated implementation, the query handler 150 includes a natural language parser 202 to parse text-based queries, such as keywords, phrases, and sentences. The parser 202 is configured to extract keywords from the query, and may utilize syntactic and semantic information from natural language queries to better understand and identify keywords. The parsed results are used as input to the semantic network that associates keywords with images in the database 142.

Fig. 3 pictorially illustrates a semantic network 300. The network defines keyword-image links that associate keywords 302(1), 302(2), ... 302(N) with images 304(1), 304(2), 304(3), ..., 304(M) in the database 142. The keyword-image links are illustrated as arrows. Weights w are assigned to each individual link to represent the degree of relevance in which a keyword describes the linked image's semantic content. For example, the first keyword 302(1) is associated with three images 304(1)-304(3) and the association with the first image 304(1) is

1 assigned a weight w_{11} , the association with the second image 304(2) is assigned a
2 weight w_{12} , and the association with the third image 304(3) is assigned a weight
3 w_{13} .

4 Keyword-image associations may not be available at the beginning.
5 However, there are several ways to obtain such associations. The first method is
6 to simply manually label images and assign strong weights to the keyword-image
7 link. This method can be expensive and time consuming.

8 To reduce the cost of manual labeling, an automatic approach may be
9 employed. One possible approach is to leverage the Internet and its countless
10 number of users by implementing a crawler that visits different websites and
11 download images. The data pertaining to the image, such as the file name and the
12 ALT tag string within the IMAGE tags of the HTML files, are saved as keywords
13 and associated with the downloaded image. Also, the link string and the title of
14 the page may be somewhat related to the image and hence used as possible
15 keywords. Weights are then assigned to these keyword-image links according to
16 their relevance. Heuristically, this information is listed in order of descending
17 relevance: (1) link string, (2) ALT tag string, (3) file name, and (4) title of the
18 page.

19 Another approach to incorporate additional keywords into the system is to
20 utilize the user's input queries. Whenever the user feeds back a set of images
21 marked as being relevant to the query, the input keywords are added into the
22 system and linked with the images in the set. In addition, since the user indicates
23 that these images are relevant, a large weight can be assigned on each of the newly
24 created links. This latter approach is described below in more detail with
25 reference to Fig. 5.

1 With reference again to Fig. 2, there may be a situation where the user does
2 not wish to enter a text query. Instead, the user is interested in selecting an
3 example image and searching for similar images. To accommodate this scenario,
4 the user interface 200 presents a set of image categories from which the user may
5 choose. Upon selection of a category, the image retrieval system returns a sample
6 set of images pertaining to the category.

7 The image retrieval system accommodates this scenario with a predefined
8 concept hierarchy 204 in query handler 150. The selected category is passed to
9 the concept hierarchy 204, which identifies first level images corresponding to the
10 category from the image database 142. From the sample images, the user can
11 identify an image as the example image. The low-level features of the example
12 image are then used to initiate a content-based image retrieval operation.

13 The feature and semantic matcher 152 identify images in image database
14 142 that have keywords associated with the user query and/or contain low-level
15 features resembling the example image. The feature and semantic matcher 152
16 includes an image feature extractor 210 that extracts low-level features from the
17 candidate images in the image database 142. Such low-level features include
18 color histogram, texture, shape, and so forth. The feature extractor 210 passes the
19 features to an image feature matcher 212 to match the low-level features of the
20 candidate images with the low-level features of the example image submitted by
21 the user. Candidate images with more similar features are assigned a higher rank.

22 For text queries, the feature and semantic matcher 152 has a semantic
23 matcher 212 to identify images with associated keywords that match the keywords
24 from the query. The semantic matcher 214 uses the semantic network to locate
25

PLF

those images with links to the search keywords. Candidate images with higher weighted links are assigned a higher rank.

A ranking module 216 ranks the images such that the highest-ranking images are returned to the user as the preferred results set. The ranking takes into account the weightings assigned to keyword-image links as well as the closeness in features between two images. The set of highest-ranked images are returned to the user interface 200 and presented to the user for consideration.

The user interface 200 allows the user to mark images as more or less relevant, or entirely irrelevant. The feedback analyzer 154 monitors this user feedback. A relevance feedback monitor 220 tracks the feedback and performs both semantic-based relevance feedback and low-level feature relevance feedback in an integrated fashion. Generally, the relevance feedback monitor 220 adjusts the weights assigned to keyword-image links to train the semantic-based retrieval model and uses query point movement or re-weighting techniques to improve the feature-based retrieval model. The feedback analyzer 154 implements a machine learning algorithm 222 to adjust the semantic network and/or images in the database according to the relevance feedback. One particular implementation of an integrated framework for semantic-based relevance feedback and feature-based relevance feedback is described below in more detail under the heading "Integrated Relevance Feedback Framework".

The image retrieval system 140 offers many advantages over conventional systems. First, it locates images using both keywords and low-level features, thereby integrating keyword-based image retrieval and content-based image retrieval. Additionally, it integrates both semantic-based relevance feedback and feature-based relevance feedback.

Image Retrieval Process

Figs. 4 and 5 show an image retrieval process implemented by the image retrieval system 140 of Fig. 2. The process entails a first phase for producing an image result set from an initial query (Fig. 4) and a second phase for refining the result set and learning from the results and user feedback (Fig. 5). In one implementation, the image retrieval process is implemented as computer executable instructions that, when executed, perform the operations illustrated as blocks in Figs. 4 and 5.

In one implementation, the process assumes that a coarse concept hierarchy of the available images exists, although this assumption is not necessary. For instance, images of people may be coarsely annotated generally as "people" and more particularly as "men" and "women". In addition, the low-level features of the images in the image database 142 may be calculated offline and correlated with the images through a data structure. This removes any potential slowdown caused by computing low-level features during the image retrieval process.

At block 402, the image retrieval system 140 receives an initial query submitted by a user via the user interface 200. Suppose the user enters a search query to locate images of "tigers". The user may enter any of the following queries:

"tigers"

"tiger pictures"

"Find pictures of tigers"

"I'm looking for images of tigers."

1
2 At block 404, the query handler 150 parses the user query to extract one or
3 more keywords. In our example, the keyword "tiger" can be extracted from
4 anyone of the queries. Other words, such as "pictures" and "images" may also be
5 extracted, but we'll focus on the keyword "tiger" for illustration purposes.

6 At block 406, the image retrieval system 140 searches the image database
7 142 to identify images annotated with the keyword "tiger". The system may also
8 simultaneously search of similar words (e.g., cat, animal, etc.). If any images in
9 the database have a link association with the keyword (i.e., the "yes" branch from
10 block 408), those images are placed into a result set (block 410). The images in
11 the result set are then ranked according to the weights assigned to the keyword-
12 image links in the semantic network (block 412). Having identified a set of
13 images that match the keyword, the features and semantic matcher 152 may also
14 attempt to find other images with similar low-level features as those in the result
15 set (block 414). Any such images are then added to the result set. The expanded
16 result set is then displayed to the user via the user interface 200 (block 416).

17 It is noted that while such additional images may resemble other images in
18 the original result set, certain images discovered via low-level feature comparison
19 may have nothing to do with the search keyword. That is, operation 414 may
20 return images that resemble the color or texture of another image with a tiger, but
21 have no trace of a tiger anywhere in the image.

22 Returning to block 408, if the initial keyword search fails to locate any
23 images (i.e., the "no" branch from block 408), the image retrieval system 140
24 retrieves images in a first level of the concept hierarchy (block 420). These
25 images may be randomly selected from one or more categories in the hierarchy.

1 The images are displayed to the user to suggest possible example images (block
2 422).

3 After the initial query, the image retrieval system 140 can use the results
4 and user feedback to refine the search and train the retrieval model. The
5 refinement and learning process is illustrated in Fig. 5.

6 At block 502, the feedback analyzer 154 monitors the user feedback to the
7 images in the result set. The user may mark or otherwise indicate one or more
8 images as relevant to the search query. This can be done, for example, through a
9 user interface mechanism in which the user evaluates each image and activates
10 (e.g., by a point-and-click operation) a positive mark or a negative mark associated
11 with the image. The positive mark indicates that the image is more relevant to the
12 search, whereas the negative mark indicates that the image is less or not relevant
13 to the search.

14 From the results, the user may see certain images that he/she deems
15 relevant to the query and select the images to produce a desired set (i.e., the "yes"
16 branch from block 504). In this situation, the keywords in the original query are
17 associated with the user-selected images and a *large* weight is assigned to the
18 association link (block 506). A large weight is assigned to the link because there
19 is a higher confidence that the search is accurate when high-level keywords are
20 used to identify images. In one implementation, the weights are additive. Thus,
21 an initial link might be assigned a value of "1" to indicate an association. If the
22 keyword is subsequently associated with the image via searching, the weight may
23 be incremented by "1", such that over time, the weight increases in strength.

24 At block 508, the similar low-level features correlated with these images
25 are reorganized to be closer together in feature space. Then, for subsequent

1 searches, the system will better understand the user's intention for certain images
2 given the same keyword.

3 If the user does not see a set of images that are relevant to the search query
4 (i.e., the "no" branch from block 504), the user may select an example image and
5 refine the search to locate other images that have similar features to those of the
6 selected image (block 510). When an example image is selected (i.e., the "yes"
7 branch from block 510), the keywords in the original query are associated with the
8 user-selected images and a *small* weight is assigned to the association link (block
9 512). A small weight is assigned to the link because there is less confidence that
10 low-level image similarities produce a result as accurate as the result produced by
11 keyword matches.

12 In response to user selection of an example image for refinement, the query
13 handler 150 attempts to find other representative images in the next level of
14 concept hierarchy (block 514). The feature and semantic matcher 152 also try to
15 locate images that have similar low-level features as the example image selected
16 by the user (block 516). The resulting set of images is then displayed to the user
17 (block 518).

18 Block 520 accounts for the situation where the original query did not return
19 any relevant images, nor did the user find an image to refine the search. In this
20 situation, the image retrieval system simply outputs images in the database one
21 page at a time to let the user browse through and select the relevant images to feed
22 back into the system.

User Interface

The image retrieval system 140 supports three modes of user interaction: keyword-based search, search by example images, and browsing the image database using a pre-defined concept hierarchy. The user interface 200 accommodates these three modes.

Fig. 6 shows an example of a query screen 600 presented by the user interface 200 for entry of an initial query. The screen display 600 presents a natural language text entry area 602 that allows user to enter keywords or phrases. After entering one or more keywords, the user actuates a button 604 that initiate the search for relevant images. Alternatively, the user can browse a pre-defined concept hierarchy by selecting one of the categories listed in section 606 of the query screen 600. The user actuates the category link to initiate a search for images within the category.

The results of the keyword or content-based image retrieval are presented in a next screen. For discussion purposes, suppose the user enters the keyword "tiger" into the text entry area 602 of query screen 600.

Fig. 7 shows an example results screen 700 presented in response to entry of the keyword "tiger". Depending on display size, one or more images are displayed in the results screen 700. Here, six images 702(1)-702(6) are displayed at one time. If there are more images than can be displayed simultaneously, navigation "Next" and "Prev" buttons 704 are presented to permit browsing to other images in the result set.

The user interface allows the user to feedback relevance information as he/she browses the images. Each image has several feedback options.. For instance, each image has a "View" link 706 that allows the user to enlarge the

1 image for better viewing. Activation of a "Similar" link 708 initiates a subsequent
2 query for images with both similar semantic content and similar low-level features
3 as the corresponding image. This refined search will be presented in the next
4 screen.

5 Furthermore, each image has both positive and negative relevance marks
6 that may be individually selected by the user. The relevance marks allow the user
7 to indicate on an image-by-image basis, which images are more relevant to the
8 search query and which are less relevant. Examples of such marks include a "+"
9 and "-" combination, or a "thumbs up" and "thumbs down", or a change in
10 background color (e.g., red means less relevant, blue means more relevant).

11 In Fig. 7, images 702(1), 702(2), and 702(5) are marked with a blue
12 background, indicating a positive match that these images do in fact represent
13 tigers. Images 702(4) and 702(6) have a red background, indicating that they do not
14 match the query "tiger". Notice closely that these images contain leopards and not
15 tigers. Finally, image 702(3) has a gradient background (neither positive nor
16 negative) and will not be considered in the relevance feedback. This image
17 presents a wolf, which has essentially no relevance to tigers.

18 After providing relevant feedback, the user activates the "Feedback" button
19 710 to submit the feedback to the feedback analyzer 154. The learning begins at
20 this point to improve the image retrieval process for future queries.

21 **Integrated Relevance Feedback Framework**

22 This section described an exemplary implementation of integrating
23 semantic-based relevance feedback with low-level feature-based relevance
24 feedback. Semantic-based relevance feedback can be performed relatively easily
25

1 compared to its low-level feature counterpart. One exemplary implementation of
2 semantic-based relevance feedback is described first, followed by how this
3 feedback can be integrated with feature-based relevance feedback.

4 For semantic-based relevance feedback, a voting scheme is used to update
5 the weights w_{ij} associated with each link in the semantic network 300 (Fig. 3).
6 The weight updating process is described below.

7
8 Step 1: Initialize all weights w_{ij} to 1. That is, every keyword is initially
9 given the same importance.

10 Step 2: Collect the user query and the positive and negative feedback
11 examples.

12 Step 3: For each keyword in the input query, check if any of them is not in
13 the keyword database. If so, add the keyword(s) into the database
14 without creating any links.

15 Step 4: For each positive example, check if any query keyword is not linked
16 to it. If so, create a link with weight "1" from each missing
17 keyword to this image. For all other keywords that are already
18 linked to this image, increment the weight by "1".

19 Step 5: For each negative example, check to see if any query keyword is
20 linked with it. If so, set the new weight $w_{ij} = w_{ij}/4$. If the weight w_{ij}
21 on any link is less than 1, delete that link.

22
23 It can be easily seen that as more queries are input, the system is able to
24 expand its vocabulary. Also, through this voting process, the keywords that
25 represent the actual semantic content of each image are assigned larger weights. It

1 should be noted, however, that the above weight update scheme is just one of
2 many reasonable ones.

3 As noted previously, the weight w_{ij} associated on each keyword-image link
4 represents the degree of relevance in which this keyword describes the linked
5 image's semantic content. For retrieval purposes, another consideration is to
6 avoid having certain keywords associated with a large number of images in the
7 database. The keywords with many links to many images should be penalized.
8 Therefore, a relevance factor r_k of the k^{th} keyword association be computed as
9 follows:

10
11
$$r_k = w_k (\log_2 \frac{M}{d_i} + 1) \quad (2)$$

12
13 where M is the total number of images in the database, $w_k = w_{mn}$ if $m = i$ and 0
14 otherwise, and d_i is the number of links that the i^{th} keyword has.

15 Now, the above semantic-based relevance feedback needs to be integrated
16 with the feature-based relevance feedback. It is known from previous research
17 (See, Rui, Y., Huang, T. S. "A Novel Relevance Feedback Technique in Image
18 Retrieval," ACM Multimedia, 1999) that the ideal query vector q_i^* for feature i is
19 the weighted average of the training samples for feature i given by:

20
21
$$q_i^{r*} = \frac{\pi^T X_i}{\sum_{n=1}^N \pi_n} \quad (3)$$

22

23
24 where X_i is the $N \times K_i$ training sample matrix for feature i , obtained by stacking the
25 N training vectors x_{ni} into a matrix, and where N is an element vector $\pi = [\pi_1, \dots, \pi_N]$

that represents the degree of relevance for each of the N input training samples.

The optimal weight matrix W_i^* is given by:

$$W_i^* = (\det(C_i))^{\frac{1}{K_i}} C_i^{-1} \quad (4)$$

where C_i is the weighted covariance matrix of X_i . That is:

$$C_{i_{rs}} = \frac{\sum_{n=1}^N \pi_n (x_{nir} - q_{ir})(x_{nis} - q_{is})}{\sum_{n=1}^N \pi_n} \quad r, s = 1, \dots, K_i \quad (5)$$

The critical inputs into the system are x_{ni} and π . Initially, the user inputs these data to the system. However, this first step can be eliminated by automatically providing the system with this initial data. This is done by searching the semantic network for keywords that appear in the input query. From these keywords, the system follows the links to obtain the set of training images (duplicate images are removed). The vectors x_{ni} can be computed easily from the training set. The degree of relevance vector π is computed as follows:

$$\pi_i = \alpha^M \sum_{j=1}^M r_{ij} \quad (6)$$

where M is the number of query keywords linked to the training image i , r_{jk} is the relevance factor of the j^{th} keyword associated with image i , and $\alpha > 1$ is a suitable constant. The degree of relevance of the i^{th} image increases exponentially with the

number of query keywords linked to it. In the one implementation, an experimentally determined setting of $\alpha = 2.5$ yielded the best results.

To incorporate the low-level feature based feedback and ranking results into high-level semantic feedback and ranking, a unified distance metric function G_j is defined to measure the relevance of any image j within the image database in terms of both semantic and low-level feature content. The function G_j is defined using a modified form of the Rocchio's formula (See Background) as follows:

750

$$G_j = \log(1 + \pi_j) D_j + \beta \left\{ \frac{1}{N_R} \sum_{i \in N_R} \left[\left(1 + \frac{I_i}{A_1} \right) S_{ij} \right] \right\} - \gamma \left\{ \frac{1}{N_N} \sum_{i \in N_N} \left[\left(1 + \frac{I_i}{A_2} \right) S_{ij} \right] \right\} \quad (7)$$

where D_j is the distance score computed by the low-level feedback, N_R and N_N are the number of positive and negative feedbacks respectively, I_i is the number of distinct keywords in common between the image j and all the positive feedback images, I_2 is the number of distinct keywords in common between the image j and all the negative feedback images, A_1 and A_2 are the total number of distinct keywords associated with all the positive and negative feedback images respectively, and finally S_{ij} is the Euclidean distance of the low-level features between the images i and j .

The first parameter α in Rocchio's formula is replaced with the logarithm of the degree of relevance of the j^{th} image. The other two parameters β and γ can be assigned a value of 1.0 for simplicity. However, other values can be given to emphasize the weighting difference between the last two terms.

Using the method described above, the combined relevance feedback is provided as follows.

1
2 Step 1: Collect the user query keywords

3 Step 2: Use the above method to compute x_{ni} and π and input them into the
4 low-level feature relevance feedback component to obtain the
5 initial query results.

6 Step 3: Collect positive and negative feedbacks from the user.

7 Step 4: Update the weighting in the semantic network according to the 5-
8 step process described earlier in this section.

9 Step 5: Update the weights of the low-level feature based component.

10 Step 6: Compute the new x_{ni} and π and input into the low-level feedback
11 component. The values of x_{ni} may be computed beforehand in a
12 pre-processing step.

13 Step 7: Compute the ranking score for each image using equation 7 and sort
14 the results.

15 Step 8: Show new results and go to step 3.

16
17 The image retrieval system is advantageous over prior art systems in that it
18 learns from the user's feedback both semantically and in a feature based manner.
19 In addition, if no semantic information is available, the process degenerates into
20 conventional feature-based relevance feedback, such as that described by Rui and
21 Huang in the above-cited "A Novel Relevance Feedback Technique in Image
22 Retrieval".
23
24
25

New Image Registration

Adding new images into the database is a very common operation under many circumstances. For retrieval systems that entirely rely on low-level image features, adding new images simply involves extracting various feature vectors for the set of new images. However, since the retrieval system utilizes keywords to represent the images' semantic contents, the semantic contents of the new images have to be labeled either manually or automatically. In this section, an automatic labeling technique is described.

The automatic labeling technique involves guessing the semantic content of new images using low-level features. The following is an exemplary process:

Step 1: For each category in the database, compute the representative feature vectors by determining the centroid of all images within this category.

Step 2: For each category in the database, find the set of representative keywords by examining the keyword association of each image in this category. The top N keywords with largest weight whose combined weight does not exceed a previously determined threshold τ are selected and added into the list the representative keywords. The value of the threshold τ is set to 40% of the total weight.

Step 3: For each new image, compare its low-level feature vectors against the representative feature vectors of each category. The images are labeled with the set of representative keywords from the closest matching category with an initial weight of 1.0 on each keyword.

1
2 Because the low-level features are not enough to present the images'
3 semantics, some or even all of the automatically labeled keywords will inevitably
4 be inaccurate. However, through user queries and feedbacks, semantically
5 accurate keywords labels will emerge while semantically inaccurate keywords will
6 slowly be eliminated.

7 Another problem related to automatic labeling of new images is the
8 automatic classification of these images into predefined categories. This problem
9 is addressed by the following process:

10
11 Step 1: Put the automatically labeled new images into a special "unknown"
12 category.

13 Step 2: At regular intervals, check every image in this category to see if any
14 keyword association has received a weight greater than a threshold
15 ξ . If so, extract the top N keywords whose combined weight does
16 not exceed the threshold τ .

17 Step 3: For each image with extracted keywords, compare the extracted
18 keywords with the list of representative keywords from each
19 category. Assign each image to the closest matching category. If
20 none of the available categories result in a meaningful match, leave
21 this image in the "unknown" category.

22
23 The keyword list comparison function used in step 3 of the above algorithm
24 can take several forms. An ideal function would take into account the semantic
25 relationship of keywords in one list with those of the other list. However, for the

1 sake of simplicity, a quick function only checks for the existence of keywords
2 from the extracted keyword list in the list of representative keywords.

3 4 Conclusion

5 Although the description above uses language that is specific to structural
6 features and/or methodological acts, it is to be understood that the invention
7 defined in the appended claims is not limited to the specific features or acts
8 described. Rather, the specific features and acts are disclosed as exemplary forms
9 of implementing the invention.
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25